Computer Security

Cryptology:
Cryptography and Cryptanalysis;
Classical Cryptosystems

Cryptology

- *Cryptology* is the scientific study of:
  - *Cryptography* -- The study of concealing information with the use of mathematical transformations; performed by *cryptographers*
  - *Cryptanalysis* -- The practice of revealing information from cryptographically hidden data, using mathematical, analytical and heuristic techniques (usually without the consent of the cryptographer); performed by *cryptanalysts*
- *Cryptology* is concerned with conveying messages from one place to another while hiding those messages from potentially prying eyes. *Plaintext* (a.k.a. *cleartext*) is the message that will be put into secret form. Usually, plaintext is in the native language of the sender/receiver.

Types of Cryptographic Attacks

- Ciphertext Only
  - Attacker knows only the ciphertext
  - Most difficult, because least amount of information available
- Known Plaintext
  - Attacker knows both the plaintext (or part of it, such as a known or repeated header) and the ciphertext -- goal is to find the key
  - More powerful than Ciphertext Only
- Chosen Plaintext
  - Attacker chooses the plaintext(s), and observes the corresponding ciphertext(s)
  - More powerful than Known Plaintext
- Chosen Ciphertext
  - Really Chosen Ciphertext and Plaintext
  - Attacker chooses both plaintext and ciphertext values
  - More powerful than Chosen Plaintext

Message Hiding Techniques

- There are two major ways by which a message can be hidden:
  - Steganography
  - Cryptography

Steganography

- Conceals the very existence of the message
  - Invisible ink
  - Microdots
  - Acrostics (An *acrostic* is a composition, usually in verse, in which sets of letters (as the initial or final letters of the lines), taken in order, form a word or phrase or a regular sequence of letters of the alphabet)
  - Hiding information in image and sound files, etc.
  - Digital watermarking (Encoding an identifying code into digitized music, video, picture, or other file is known as a digital watermark.)

"Steganography (literally meaning *covered writing*) dates back to ancient Greece, where common practices consisted of etching messages in wooden tablets and covering them with wax, and tattooing a shaved messenger's head, letting his hair grow back, then shaving it again when he arrived at his contact point."

`http://www.webopedia.com/TERM/S/steganography.html`

Steganography

- Can you find the secret message hidden in the following text?

Because there are many unscrupulous individuals and organizations out there, security is paramount. Regular security checks are strongly encouraged. You would be well advised to make them part of your daily routine. Aggressive application of security techniques is called for in all circumstances. Not applying these techniques can be the source of serious problems.

Cryptography

- Does not conceal the presence of the message
- Renders the contents of a message unintelligible to outsiders
- There are two major methods of cryptography:
  - *Codes*
    - Usually map words to words, so depend on the language being used.
    - Encoding is through a large reference, called a code-book, which specifies how the mapping is done.
  - *Ciphers*
    - Usually transform units of a fixed length, by means of some mathematical function.
    - Can be relatively independent of the language being used in the message

Simple Example of a Code

Word	Codeword
…	…
At	Hot
Attack	I
…	…
…	…
Dawn	Curry
…	…
Enemy	Like
…	…

- Using the code-book to the left, we would encode:

`Attack enemy at dawn`

as:

`I like hot curry`

Ciphers

- Early ciphers were based on mapping alphabetic letters to other letters in the alphabet.
- The simplest ciphers can be broken down into:
  - Transposition ciphers
  - Substitution ciphers

Transposition Ciphers

- A *Transposition Cipher* merely rearranges the letters in the plaintext to form the ciphertext. The letters in the plaintext are not changed.
  - One type of transposition cipher is the *rail fence* cipher, where the plaintext is written in two rows, first down, then across. For example, rewriting the plaintext "WITHDRAW TROOPS" (and ignoring spaces), we get:

    `WTDATOP`
    `IHRWROS`

    which results in the ciphertext **`WTDATOPIHRWROS`**

Breaking a Transposition Cipher

- It is easy to cryptanalyze a transposition cipher, by *anagramming*:
  - Using tables of n-gram frequencies to identify common n-grams.
  - In the simple rail fence example, we would identify *digrams* ("2-grams") by looking at the relative frequencies of 2-letter occurrences in English. Eventually, we would figure out the original plaintext.

Substitution Ciphers

- *Substitution Ciphers* change characters in the plaintext, to produce the ciphertext
  - The substitution constitutes a 1-to-1 mapping of plaintext letter to ciphertext letter.
  - In other words, a given letter in the plaintext always maps to the same letter in the ciphertext.

Shift Ciphers

- The simplest form of Substitution Cipher is a *Shift Cipher*:
  - Each letter in the plaintext maps to a letter in ciphertext, where the ciphertext letter is at a fixed offset position in the alphabet from the original plaintext letter.

The Caesar Cipher

- Julius Caesar (Roman Emperor, 100 - 44 B.C.) needed to send messages to his generals, but he did not trust the messengers (or perhaps he thought that they might be intercepted), so he used a very simple cipher, now known as the Caesar Cipher. He took every character in the message and *shifted* it three letters down in the alphabet (or course, he used the Latin alphabet, but we'll use our own alphabet here):

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C

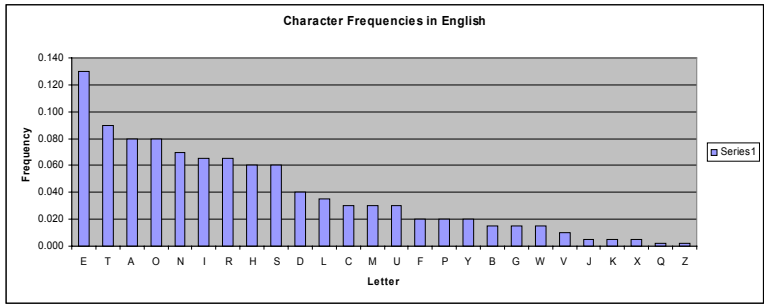
So, a message "**`Veni, vidi, vici`**" would translate to:

`Sbkf, sfaf, sfzf`

For example, see this Java Applet.

The Caesar Cipher

- Unfortunately, the Caesar Cipher (even with shifts other than 3) is easily breakable.
  - It is subject to a letter frequency analysis.
    - In English, here are the relative frequencies of letters in typical text:



Breaking a Shift Cipher

- Given a large enough sample of ciphertext, and using a letter frequency analysis, it is relatively easy to determine the corresponding plaintext:
  - Compute the relative frequency of each letter in the ciphertext
  - Let $\phi(i)$ be the correlation of the frequency of each letter in the ciphertext with the character frequencies in English:

$$\phi(i) = \sum_{c=0}^{25} f(c)\,p(c-i)$$

where $f(c)$ is the frequency of the character c (expressed as a fraction) and $p(c - i)$ is the frequency of the letter (c - i) in English.

  - The correlation, $\phi(i)$, should be a maximum when the the key (shift) translates the ciphertext into English.

Breaking a Shift Cipher

- Given computers, for a simple shift cipher, it's even easier.
  - There are only 26 possible shift keys:
    - The key (shift) can be any value in the range 0 through 25 (there are 26 letters in the English alphabet), we can simply display all the 26 possible cases, and (we hope!) easily pick out the one that gives plaintext.

Breaking a Shift Cipher

- Assume you're trying to decipher:

  "Xawi ia ql, Oykppu!"

- It's easy to list all the 26 possibilities:

```
Xawi ia ql, Oykppu!        Knjv vn dy, Blxcch!
Wzvh hz pk, Nxjoot!        Jmiu um cx, Akwbbg!
Vyug gy oj, Mwinns!        Ilht tl bw, Zjvaaf!
Uxtf fx ni, Lvhmmr!        Hkgs sk av, Yiuzze!
Twse ew mh, Kugllq!        Gjfr rj zu, Xhtyyd!
Svrd dv lg, Jtfkkp!        Fieq qi yt, Wgsxxc!
Ruqc cu kf, Isejjo!        Ehdp ph xs, Vfrwwb!
Qtpb bt je, Hrdiin!        Dgco og wr, Ueqvva!
Psoa as id, Gqchhm!        Cfbn nf vq, Tdpuuz!
Ornz zr hc, Fpbggl!        Beam me up, Scotty!
Nqmy yq gb, Eoaffk!        Adzl ld to, Rbnssx!
Mplx xp fa, Dnzeej!        Zcyk kc sn, Qamrrw!
Lokw wo ez, Cmyddi!        Ybxj jb rm, Pzlqqv!
```

For example, see this Java Applet.

More Complex Substitution Ciphers

- Of course, it's not necessary to restrict ourselves to a simple shift; we can map the plaintext letters to the ciphertext letters using any *permutation* of the alphabet.
  - This gives us, not 26 possibilities, but 26! possibilities:

    $26! = 403291461126605635584000000 \approx 4 \times 10^{26}$ possible keys

    which is a smidgen more than 26 different keys!
    - If we tried 1 microsecond to try each key, to try them all it would take:

      $4 \times 10^{26} / (10^6 \times 60 \times 60 \times 24 \times 365) = 1.7 \times 10^{13} \approx 17$ trillion years!
- This type of cipher is called a *MonoAlphabetic Substitution Cipher*, because a single cipher alphabet (i.e. a mapping from plaintext to ciphertext) is used per message.
  - A shift cipher is merely a special case of this kind of cipher.

For example, see this Java Applet.

MonoAlphabetic Substitution Ciphers

- You would think that, with this number of key possibilities, that the cipher would be pretty secure, right?
  - Unfortunately, monoalphabetic substitution ciphers are also subject to a letter frequency analysis...

Breaking a MonoAlphabetic Substitution Cipher

- First, determine the relative frequency of the letters in the ciphertext, and compare them with the known frequency distribution of letters in the English language.
  - Depending on the amount of ciphertext, and how well its original plaintext represents English letter frequencies (there are always some statistical variations), this can lead you to some, possibly provisional, letter mapping assignments.
  - You could then start to fill in the plaintext to see whether it looks like a skeleton of something recognizable.
  - You could also look for other regularities; certain words (like '`the`', '`a`', '`if`', '`of`', etc.) might be known, or guessed at, based on repeating sequences of ciphertext letters. This is more apparent if spaces in the plaintext are preserved in the ciphertext (which is why spaces are often removed)

Breaking a MonoAlphabetic Substitution Cipher

- You could also look for the frequency of 2-letter combinations, known as *digrams*. For example, the most common English digram is '`th`'
  - For a particular ciphertext, we might observe that the most common digram is ZW, so we could try mapping Z to '`t`' and W to '`h`'.
  - Assume that we see a 3-letter sequence ZWP; this might correspond to plaintext '`the`', so we would then try mapping P to '`e`'; perhaps P is high on the list of frequent characters in the ciphertext.
  - We might now see a four letter sequence ZWSZ in the ciphertext, and with our previous assumptions/guesses, this is of the form '`th?t`"; most likely, this corresponds to plaintext '`that`'.
  - Following similar analyses and some trial and error, it is often possible (or even easy) to break the ciphertext.

Here's a potentially useful web site which lists English letter and word frequencies.

Polygram Substitution Ciphers

- Mapping single letters to single letters is not secure, so cryptographers came up with the concept of mapping *entire blocks* of plaintext letters to blocks of ciphertext letters.
  - For example, using a block size of 8, we could map blocks of 8 letters at a time: `AAAAAAAA` through `ZZZZZZZZ` -- there are 26^8 distinct possibilities.
  - To break such a cipher, you would have to have a table of size 26^8 = 208,827,064,576 blocks, and also know the relative frequencies of the occurrence of 8-letter blocks in the plaintext.

The Playfair Cipher

- In 1854, Sir Charles Wheatstone invented the *Playfair Cipher*, which is a polygram substitution cipher using a block size of 2 (which is not very secure)
  - Based on the use of a 5×5 square matrix of letters, constructed starting from a keyword or keyphrase. Each unique letter from the phrase is inserted into the square, until there are no more letters, and then the remaining letters of the alphabet are added to fill the square.
  - For example, the phrase "Cynicism is the last refuge of the romantic" produces the matrix:

C	Y	N	I/J	S
M	T	H	E	L
A	R	F	U	G
O	B	D	K	P
Q	V	W	X	Z

The Playfair Cipher

- Here are the rules to encipher a piece of plaintext:

  `Massachusetts goes Republican!`

  – First, eliminate all non-letter characters, and upcase all letters:

  `MASSACHUSETTSGOESREPUBLICAN`

  – Then, arrange the plaintext in pairs of letters. If any pair of letters contains the same letter (for example, 'SS'), then insert an 'X':

  `MA SX SA CH US ET TS GO ES RE PU BL IC AN`

  If there is a last character not paired, add an 'X'.

The Playfair Cipher

- For each pair of plaintext characters, call the first `p`, and the second `q`; the corresponding ciphertext characters `c` and `d`:
  – If `p` and `q` are in the *same row* of the matrix, `c` is the letter *to the right of* `p`, and `d` is the letter *to the right of* `q`, wrapping around if necessary
  – If `p` and `q` are in the *same column* of the matrix, `c` is the letter *below* `p`, and `d` is the letter *below* `q`, wrapping around if necessary
  – If `p` and `q` share neither the same row nor column, they define the corners of a square. The other two corners of the square are `c` and `d`, with `c` being the letter in the same column as `p`.

MA SX SA CH US ET TS GO ES RE PU BL IC AN
AO ZI GC MN IG LH YL PA IL TU GK TP SY CF

C	Y	N	I/J	S
M	T	H	E	L
A	R	F	U	G
O	B	D	K	P
Q	V	W	X	Z

How would you decipher this message?

Try a Playfair Cipher Java applet

PolyAlphabetic Substitution Ciphers

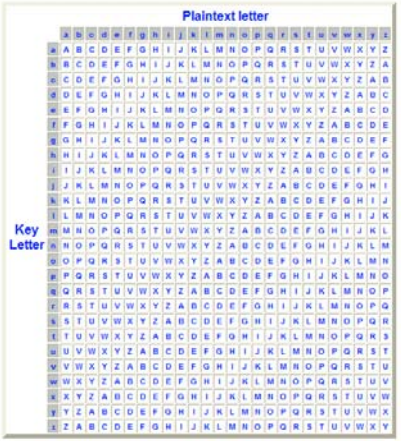
- Because monoalphabetic substitution ciphers are so notoriously insecure, cryptographers invented *PolyAlphabetic Substitution Ciphers*

- A PolyAlphabetic Substitution Cipher has:
  – A set of related monoalphabetic substitution rules, and
  – A key to determine which particular rule is chosen for a given transformation

The Vigenère Cipher

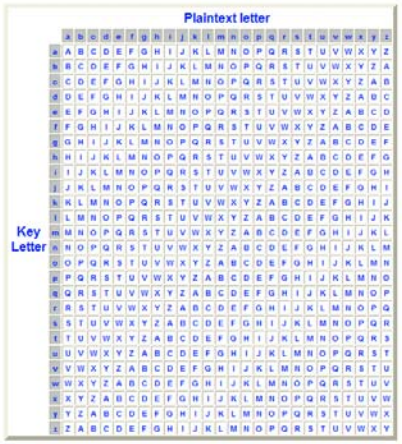
- The best known (and one of the simplest) polyalphabetic substitution cipher is the *Vigenère Cipher*

  It uses a *Vigenère Tableau* (table in French) or *Vigenère Square*:



The Vigenère Cipher

- To encrypt a plaintext message:
  - Choose a key.
  - Extract the first letter in the plaintext, p, and the first letter in the key, q
  - Use p to select a column in the tableau and q to select a row in the tableau. The character in the corresponding cel is the ciphertext character
  - Repeat for the second plaintext character, and second key letter, and so on. When you come to the end of the key, you wrap around to the first letter of the key.
  - The length of the key is called the *period of the cipher.*



How would you decipher the resulting ciphertext?

The Vigenère Cipher

- The strength of this cipher is that there are multiple ciphertext letters for each plaintext letter, and so the letter frequency information is obscured.
- For a long time, the Vigenère Cipher was considered unbreakable.
  - Then a retired Prussian cavalry officer named Kasiski noticed that repetitions occur in the ciphertext when characters of the key appear over the same characters in the ciphertext. The number of characters between the repetitions is a multiple of the period.
  - The longer the period, the more secure is the cipher -- preferably the key value should be chosen to be as long as the plaintext, and should have no statistical relationship with it.

The Vigenère Cipher

- Variations of the Vigenère Cipher were introduced:
  - The *Full Vigenère Cipher*
    - Use of a tableau with each line representing a permutation of the alphabet, not just a simple shift
  - The *Auto-Key Vigenère Cipher*
    - Both the key and [part of] the plaintext are the used as the real key
  - The *Running Key Vigenère Cipher (Vernam Cipher)*
    - Makes use of a very long key -- for example, a passage from a book, or a running loop of tape.

but each one of them is still vulnerable to a letter frequency analysis.

The One-Time Pad

- A U.S. Army Signal Officer, Joseph Mauborgne, proposed an improvement on the Vernam Cipher -- the *One-Time Pad.*
  - Uses a random key that is truly as long as the message, with no repetitions.
  - This type of cipher is *provably unbreakable*.
  - It produces random output that bears no statistical relationship to the plaintext, and so there is no way to break the cipher.
- In practice, the one-time pad has problems:
  - No practical way of making large quantities of random keys.
  - Key distribution is a truly daunting task.
  - For these reasons, the one-time pad is not used today

Homophonic Substitution Ciphers

- One approach to thwarting letter frequency analysis cipher attacks was the use of homophones*
  - Letters which occurred more frequently in the language were given multiple choices of ciphertext symbols.
  - Each letter of the plaintext alphabet was allocated a number of 2-letter ciphertext translations; the number was roughly proportional to the frequency of occurrence of the letter in typical plaintext.
  - Each translation for a given letter should be chosen as randomly as possible.
  - The resulting ciphertext is larger than the corresponding plaintext.
  - Were effective, but not used much now because of dependency on the language, and the larger ciphertext size.

***ho·mo·phone**
1 : one of two or more words pronounced alike but different in meaning or derivation or spelling (as the words *to*, *too*, and *two*)
2 : a character or group of characters pronounced the same as another character or group

Combining Substitution and Transposition Ciphers

- You can, of course, combine different cipher techniques to produce a (hopefully) more secure cipher.
  - Typically, you would have multiple stages, each using a different technique, and working on the output of the previous stage.
  - The German ADFGVX cipher used such techniques. It was a surprisingly simple cipher, but was only broken after considerable time and effort.

Summary

- We defined some cryptographic terms and concepts
- We reviewed some general principles of cryptography
- We took a look at a representative set of classical ciphers
- Soon, we will go further, and learn more about modern day cryptographic ciphers.